

# Objective C Programming

## Lecture 6

Key-value observing. NSObjectController. iOS example.

International University of Information Technology  
Almaty, 2012

# Project 2 – 4

- Designing and building your app
- Project 2: description, requirements, interface
- Project 3: building
- Project 4: testing, presenting

# Project 2 – 4

- OS X or iOS
- Must be executable – project 4 will be marked 0 unless your app runs
- English language (unless other languages is required by your app)
- We will talk about publishing your apps on the appStore

# A bit more of Foundation

- Fast Enumeration
- Dictionaries
- Sets

# Fast Enumeration

- Works for classes that conform to `NSFastEnumeration` protocol

```
for ( Type newVariable in expression ) { statements }
```

```
NSArray *array = [NSArray arrayWithObjects:  
    @"one", @"two", @"three", @"four", nil];  
  
for (NSString *element in array) {  
    NSLog(@"element: %@", element);  
}
```

# Dictionaries

```
NSDictionary *myLibrary = [NSDictionary dictionaryWithObjectsAndKeys:  
    @"Wind in the Willows", @"100-432112",  
    @"Tale of Two Cities ", @"200-532874",  
    @"Sense and Sensibility", @"202-546549",  
    @"Shutter Island", @"104-109834",  
    nil];
```

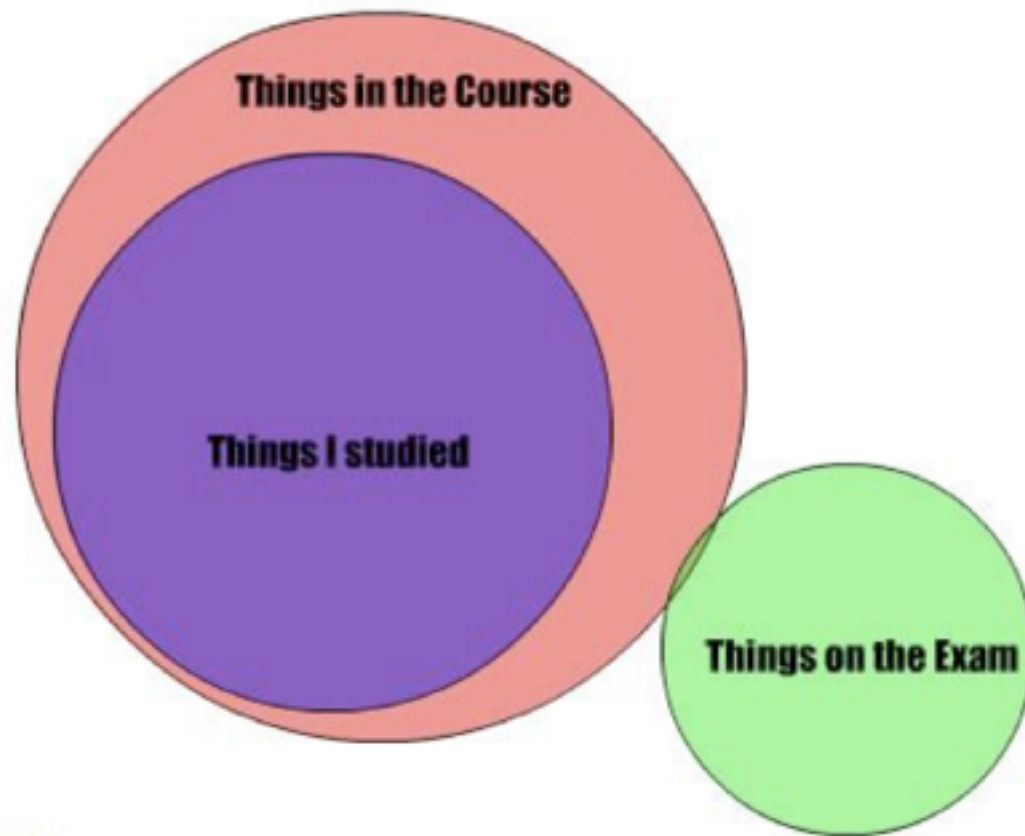
```
NSMutableDictionary *myLibrary = [NSMutableDictionary dictionary];
```

```
[myLibrary setObject: @"Wind in the Willows" forKey: @"100-432112"];  
[myLibrary setObject: @"Tale of Two Cities " forKey: @"200-532874"];  
[myLibrary setObject: @"Sense and Sensibility" forKey: @"202-546549"];  
[myLibrary setObject: @"Shutter Island" forKey: @"104-109834"];
```

# Working with dictionaries

- [myDict objectForKey: ...]
- [myDict removeObjectForKey: ...]
- [myDict removeAllObjects]

## Final Exams

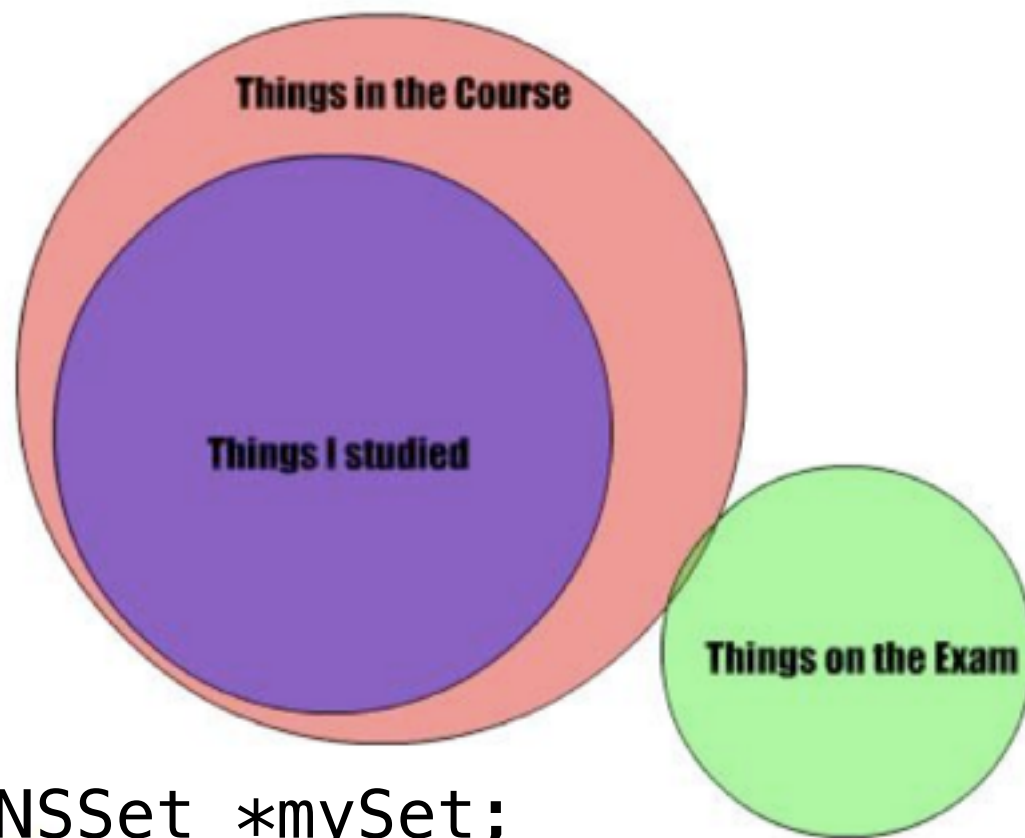


# Sets

10/10/12



## Final Exams



# Sets

```
NSSet *mySet;  
NSSet *yourSet;  
mySet = [NSSet initWithObjects:@"someString", 45, someObject, nil];  
yourSet = [NSSet initWithObjects:@"blabla", 3, 45, nil];
```

```
NSLog(@"%@", [mySet count]);  
BOOL x = [mySet isSubsetOfSet:yourSet]; // false  
BOOL y = [mySet isEqualToSet:yourSet]; // false  
BOOL z = [mySet intersectsSet: yourSet]; // true  
id m = [mySet member:@"someString"]; // returns @"someString"
```

Next – module II

# Key-Value coding

- Key-value coding (or KVC) is a mechanism that allows you to set and get the value of a variable by its name.

```
@interface Student : NSObject
{
    NSString *firstName;
}
@end
```

```
Student *s = [[Student alloc] init];
[s setValue:@"Larry" forKey:@"firstName"];
NSString *x = [s valueForKey:@"firstName"];
```

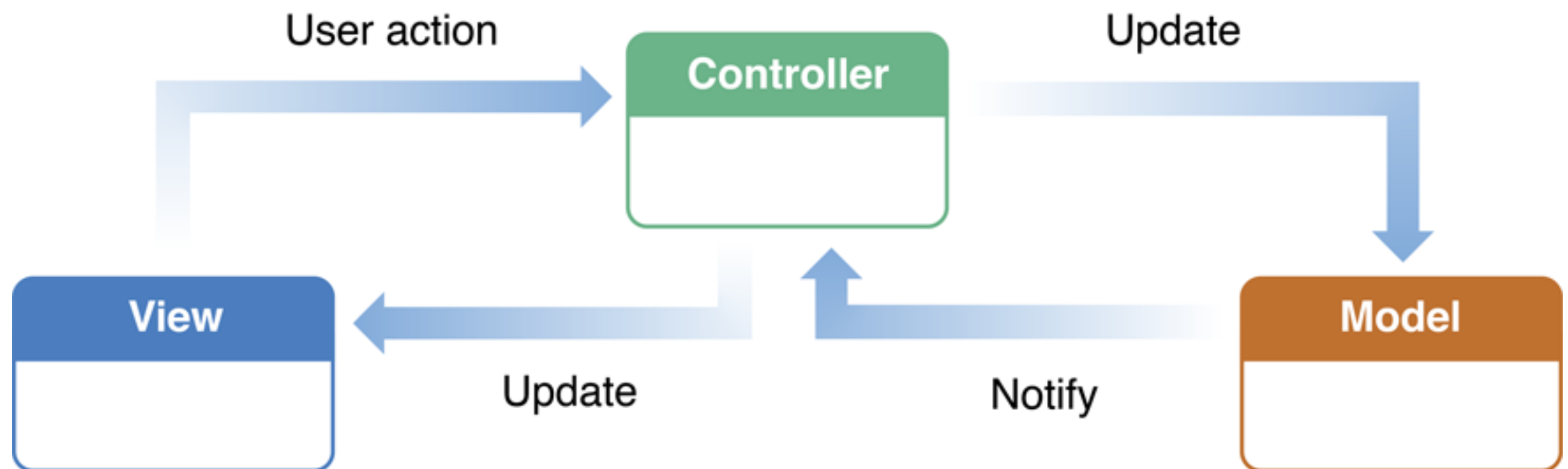
# Bindings

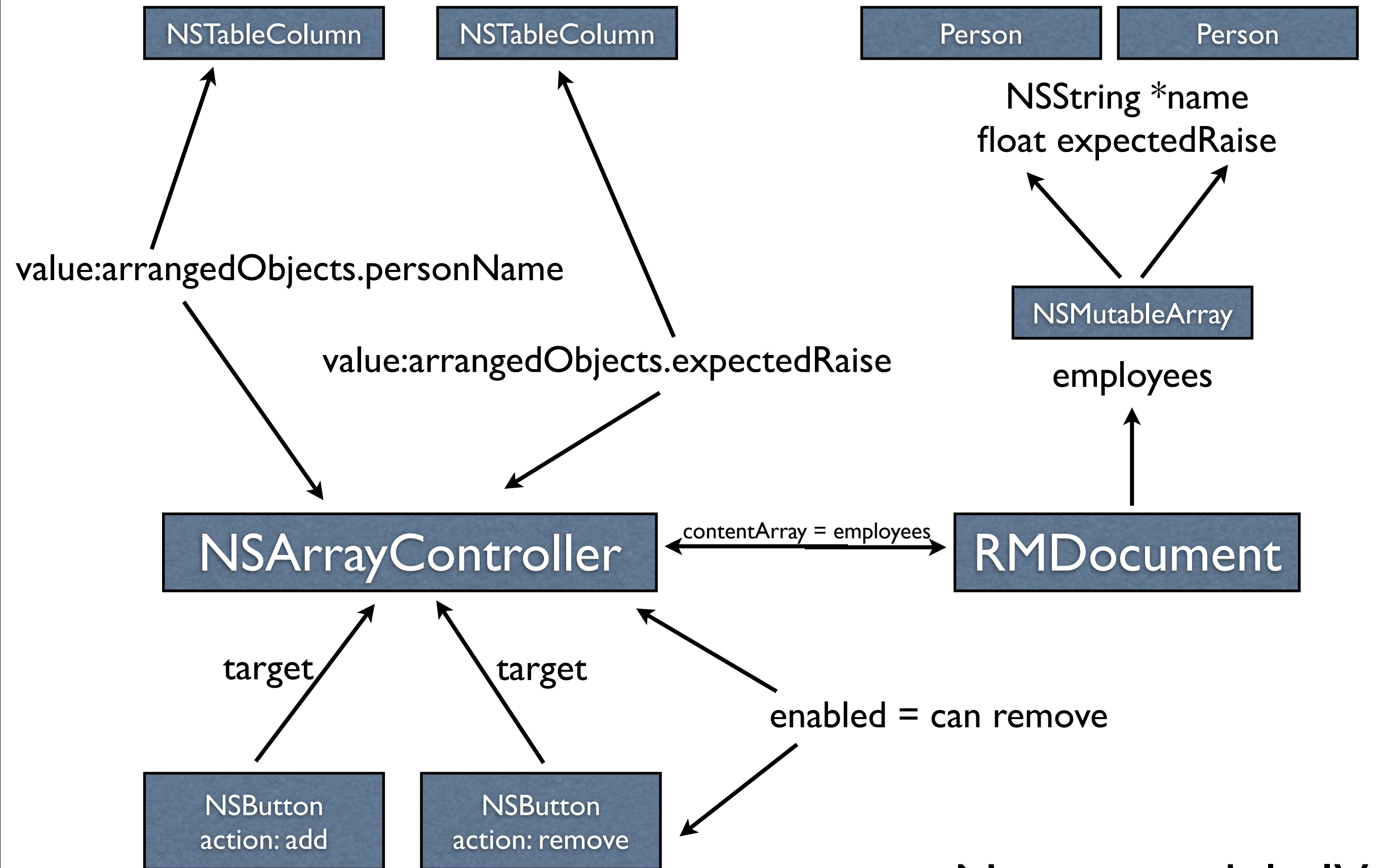
## Key-value observing

- Let's code something

Next – module III

# NSController





Next – module IV

# iOS app example